Radix sort
(Bucket sort)

IDEAS2IT

Radix sort (Bucket sort)

- ➔ It is a non-comparison based sorting algorithm, where it does grouping by the number place and position.
- ➔ Sorting starts from least significant bit and ends at greatest significant bit

ALGORITHM

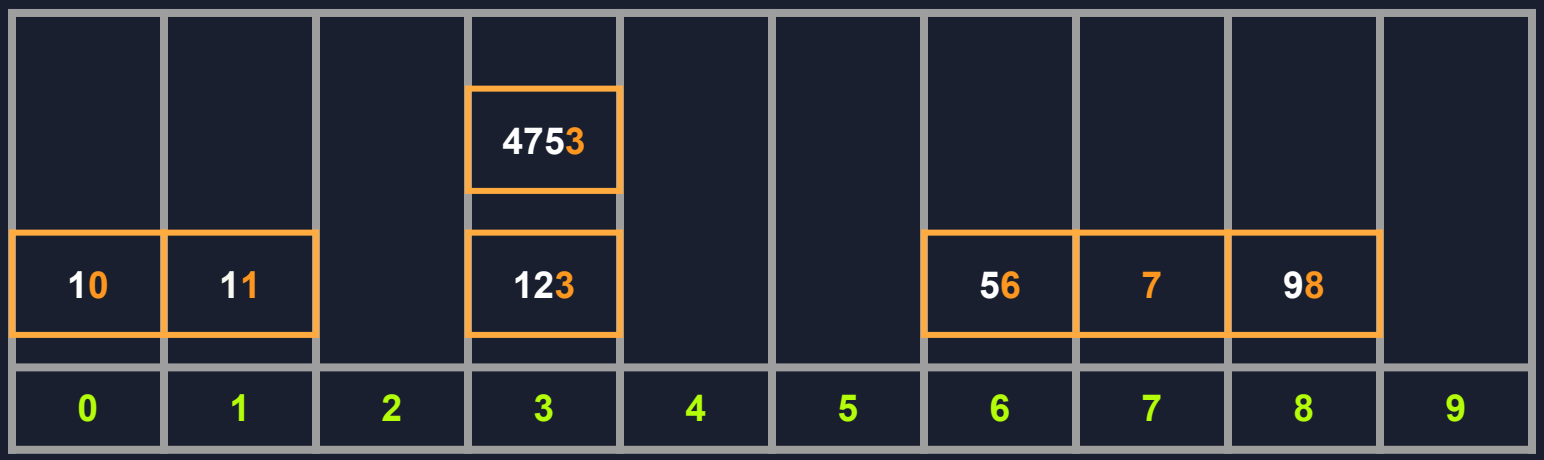
```
function radixSort(arr) {
  // length of the max digit in the array
  const max = Math.max(...arr).toString().length;

  // Runs the loop till the length of max value
  for (let i = 0; i < max; i++) {
    let buckets = Array.from({ length: 10 }, () =>  []);

    for (let j = 0; j < arr.length; j++) {
    // pushing into buckets
    buckets[Math.floor(Math.abs(arr[j]) /
     Math.pow(10, i)) % 10].push(arr[j]);
    }
    arr = [].concat(...buckets);
  }
  return arr;
}
```

EXAMPLE

➔ Sorting of array [11, 123, 10, 4753, 7, 56, 98]

Step 1: [11, 123, 10, 4753, 7, 56, 98]

➔ Place the number into the corresponding bucket based on the 1's place of number



BUCKETS

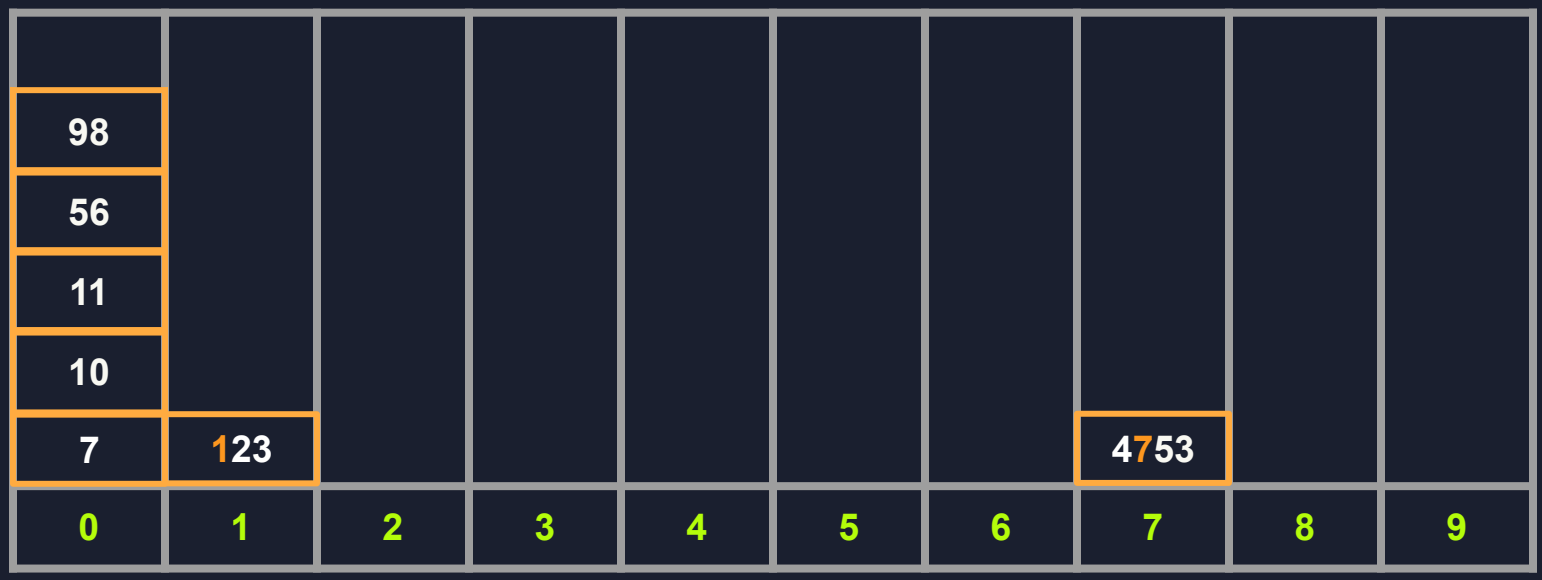
Step 2: [10, 11, 123, 4753, 56, 7, 98]

➔ Place the number into the corresponding bucket based on the 10's place of number

BUCKETS

Step 3: [7, 10, 11, 123, 4753, 56, 98]

➔ Place the number into the corresponding bucket based on the 100's place of number



Step 4: [7, 10, 11, 56, 98, 123, 4753]

➔ Place the number into the corresponding bucket based on the 1000's place of number

RESULT : [7, 10, 11, 56, 98, 123, 4753]